Volume 14 Number 1 1986 Nucleic Acids Research

Improved programs for DNA and protein sequence analysis on the IBM personal computer and other standard computer systems

David W.Mount
Department of Molecular and Cellular Biology, Biosciences West, University of Arizona, Tucson, AZ 85721, USA, and

Bruce Conrad
Department of Mathematical Sciences, The University of Lethbridge, Lethbridge, Alberta, T1K 3M4, Canada

Received 8 July 1985

ABSTRACT

We have previously described programs for a variety of types of sequence analysis (1-4). These programs have now been integrated into a single package. They are written in the standard C programming language and run on virtually any computer system with a C compiler, such as the IBM/PC and other computers running under the MS/DOS and UNIX operating systems. The programs are widely distributed and may be obtained from the authors as described below[1].

INTRODUCTION

Several reviews on the use of computers in storage and analysis of nucleic acid and protein sequences have appeared (5-11). Other reviews have discussed the software that is available from commercial and noncommercial sources (8-11). This laboratory has been interested in the application of computer sequence analysis to problems in molecular genetics. As a result, several types of computer programs for genetic experiments and for analysis of mutant sequences have been developed (1-4). These programs were designed with four features in mind: (1) portability to other computer systems through the use of a standard computer language and the most widely-used computer operating systems; (2) free availability to colleagues at other research laboratories; (3) regularly scheduled improvements and enhancements; and, (4) ease of use and excellent documentation.

PORTABILITY OF PROGRAMS

Using the same source code, the programs have been compiled and linked with the Unix C compiler to run under the UNIX operating system and with the Lattice C compiler to run under the MS/DOS operating system on a

[1] A letter requesting the programs should be sent to D.M. A standard IBM microcomputer disket containing executable forms of the program will be sent for copying and returning.

microcomputer. Earlier versions compiled with the BDS C compiler are available for CP/M80 computers. To run current versions, a microcomputer must be running under MS/DOS. The programs have been used without change on IBM/PC, IBM/XT, IBM/AT, NCR, DEC, NEC, and Otrona personal computers. Sequences up to 1 million residues long can be analyzed under MS/DOS, depending upon available memory. Pattern searching, translation and other routine tasks on long sequences such as those of phage lambda (48 kb) and Epstein-Barr virus (175 kb) are routinely performed on microcomputers with 512 kbytes of memory. At least 256 kbytes of memory is recommended for most applications.

USE OF MENU SYSTEM FOR SEQUENCE ANALYSIS PROGRAMS

Most of the analysis programs have been lumped together into one large program, called DM (DNA MENUS), from which individual routines can be reached through a system of menus, as shown in Table 1. The menu system has been used because the total length of the programs is large and because not all of the programs have to be in memory at the same time. In the compiled MS/DOS version, the individual programs are loaded from the disket into memory in response to user requests. The new program segment overlays any other segment that was previously there. The first part of the program to be loaded contains the main menu leading to all other menus and prompts. At this time, an attempt is made to read a file called sequence.dat which contains information about the length of the DNA sequence, the number of pattern matches that are likely and the expected number of open reading frames (orf's). A standard set of values is provided which should be appropriate for most applications on a machine with 256 kbytes of memory.

The major types of analysis that can be reached through the menu system are shown in Table 1. This table displays all the menu options of the programs, and illustrates the types of analysis that can be performed. These options are discussed below.

SEQUENCE LOAD FROM DISKFILE.

This option [S] in the main menu (Table 1) loads a new sequence into memory which replaces any that may have been there before. The new sequence option is only available from the main menu so that upon completion of the analysis of each sequence, it is necessary to return to the main menu to bring another into memory from a disk file. The purpose of this procedure is to avoid repetitive loading of long sequences and to increase the speed of analysis of

Table 1. Menu system used in main sequence analysis programs

```
MAIN MENU (Vers. 4.0)              OUTPUT OPTIONS

S. Sequence load from disk         1. Printer OFF (on)
P. Pattern matching                2. Terminal output continuous YES  (no)
T. Translation                     3. Output to disk file YES (no)
C. Nucleotide frequency            4. Terminal output when saving disk
O. Output options                     file YES (no)
E. Edit sequence                   5. Color bases (IBM color terminal)
X. Program exit                       YES (no)
                                   M. Return to main menu
PATTERN MATCHING MENU

1. match table 'patterns'          TRANSLATION MENU
2. match your table
3. match patterns to be typed      1. Scan sequence in one or more frames
   from keyboard                   2. Find open reading frames
4. match table 'four.cut'          3. Make disk file of translation product
5. match table 'five.cut'          4. Back translate protein file
6. match table 'six.cut'              to ambiguous DNA sequence
7. match table 'total.cut'         5. Show plot of translation product
8. approximate match to probe      6. Change three letter amino acid code
   sequence file                      to one letter code
M. Return to main menu             7. Change one letter amino acid code
                                   M. Back to main menu

PATTERN OUTPUT OPTIONS                            CODON USAGE

1. Show only matches                              1. Nucleotide frequency
2. Show entire sequence with patterns             2. Dinucleotide frequency
   matched and translation in 3 frames            3. Codon usage
3. Show only tables
4. Show fragments sizes

EDIT SEQUENCE MENU

1. Display DNA sequence file with numbers
2. Remove fragment from existing DNA sequence
   file to new disk file
3. make new sequence file from existing
   DNA sequence file with deletion
4. splice DNA sequences in two files
   and place sequence in a new file
5. make new sequence file containing
   complementary strand to another sequence
6. search for homology between 2 sequences
7. circularize sequence by adding 2 at end
8. open circular sequence at new location
M. Return to main menu
```

sequences by having the entire sequence available in memory. All the programs access the loaded sequence except the editing programs, which erase this sequence in order to manipulate other sequence files.

__Sequence Disk File__. The programs presently assume that only one sequence is present in each disk file, although this requirement can be easily changed to accommodate other types of format. After a sequence filename has been entered, the entire DNA sequence is read into memory from the disk sequence file. The user is prompted for a second entry if the file is not present on the disk. As the sequence is read, it is checked for adherence to a standard list of base symbols (Table 2) and any deviations are reported on the terminal. All symbols are read into memory and are used in the subsequent analysis.

Analysis of Parts of Sequences.

When running the individual sequence analysis programs, portions of sequences can be analyzed. The operator is asked if the entire sequence to be analyzed. If only a portion of a sequence is required, a 'no' response is entered. A prompt for the extent of the sequence to be analyzed then appears. Two numbers representing the first base in the sequence to be analyzed and the second are entered. These numbers should be separated by one or more spaces and the second is followed by a carriage return. Most of the programs require that the first number entered be smaller. A number longer than the sequence length will not be accepted nor the number zero.

Analysis of Complementary Strand.

Some of the sequence analysis programs (such as Translation options 1 and 5, and the Codon usage program - see Table 1) allow immediate analysis of the complement of a DNA sequence already loaded into memory from a disk file. In other cases, a disk file containing the complementary strand is first made using option 5, edit menu. Then, this new sequence is loaded into memory using option S, main menu, for subsequent analysis.

DNA Sequence Format.

The programs accept sequences in a very free format. Any line with ; or > in the first column is ignored by the program so that such lines may be used for comments. A variety of base symbols is recognized, as shown in Table 2. Text entered by editors that set the parity bit is acceptable. By convention, the sequence should read 5' to 3'. Spaces and tabs may be placed anywhere in the sequence and lines may be any length. The sequence should end with a .C or 2 (for circular) or .L or 1 (for linear) sequence. Otherwise, the program assumes a linear sequence. To make the complementary strand (Table 1, menu E, option 5), a .C or .L must be present. Very long sequences can be handled by most of the programs (the maximum length for MS/DOS on the IBM/PC is about 480,000 bases for 640K memory). An example of a sequence disk file is shown in Table 3.

Table 2. Ambiguous base symbols recognized by programs.

Ambiguous symbol	Bases recognized
A	A
C	C
G	G
U or T	T
P	A OR G
Y	C OR T
W	A OR T
S	G OR C
M	A OR C
K	G OR T
H	A OR T OR C
B	G OR C OR T
V	A OR C OR G
D	A OR G OR T
N OR -	A OR G OR C OR T
+	A OR G OR C OR T OR NO BASE

The pattern matching program also recognizes the following ambiguous base triplets in ambiguous DNA sequences created by back translation of protein to DNA sequences. Translation option 4 will create a file with these substitutions using as input a file containing a protein sequence in 3-letter amino acid code.

ZZZ	TAP or TGA (Termination codons)
LLL	CTN or TTP (Leu codons)
EEE	TCN or AGY (Ser codons)
RRR	CGN and AGP (Arg codons)

With the exception of the 3-letter codes for arg, leu, ser and termination codons, which are ours, these symbols are as recommended by the Nomenclature Committee of the International Union of Biochemistry. All the symbols shown may be in a DNA sequence to be analyzed. They are recognized by all of the programs and interpreted, if possible. With the exception of the ambiguous triplets, the same symbols may also be in patterns to be searched.

DATA OUTPUT OPTIONS

Prior to analysis of sequences, it is possible to select one of several types of data output by using option 0, main menu. In the output options menu, printer output (option 1) is sent first to a disk file and then printed

Table 3. Example of DNA sequence file

```
;this is a comment line
>another comment line
;this is a test linear sequence
AAA GGG TTT CCC
ACTG
A   C   T   G
1
```

later, using a text editor, because this procedure is faster and much more flexible. Option 2 limits output to no more than 10 to 24 lines of data when in 'no' status in order for the operator to read the screen before proceeding further. Paused data output also provides an opportunity to stop the analysis if a special character (control-A) is typed, followed by a carriage return. The program then returns to the last menu. When option 2 is in the 'yes' mode, data output is continuous. Usually, data will be sent to a disk file (option 3). The name test.dat is used for the disk filename if a carriage return is entered in response to the prompt for a filename. Terminal output can also be turned off (option 4) for faster output of data to a disk file. In this mode, the output is continuous until the analysis has been completed. Bases can be colored on an IBM color monitor (option 5) but output is slowed about 4-fold. To use color on the IBM/PC, the setup file config.sys should include the statement device=ansi.sys.

PATTERN MATCHING OPTIONS

A fast method of searching for predefined patterns in a DNA sequence using a nondeterministic finite automaton and regular expression matching is available in these programs (1). Using this method, a sequence about 5000 residues long can be searched for about 100 patterns, each 10 residues long, in less than 2 minutes. Patterns such as sites recognized by restriction endonucleases or regulatory sites can be found by this method. This program is reached from the option P, main menu. Several tables of patterns recognized by restriction endonucleases are provided, along with information for calculating fragment lengths (options 1 and 4-7, pattern matching menu). Patterns may be pre-entered in a file (option 2) or may be typed in on the terminal as the program is running (option 3).

Pattern Matching Tables. An example of a pattern matching table is shown in Table 4. Such a table is prepared using a text editor prior to running the pattern matching program. Tables must have the following format. As in DNA sequences, any line with ; or > in the first column is treated as a comment. Each pattern to be matched is entered on a separate line having three 'fields'. Up to the first 25 columns is reserved for an identifying name e.g. EcoRI(5'AATTC). Any character or number can be used in the pattern name, but no spaces are allowed within it, and the pattern name must start in the first column. The initial field is then separated by one or more spaces from the next field, which contains the pattern to be matched, e.g. GAATTC, which can also be up to 25 characters long. The pattern may include any of

Table 4. Example of Pattern Matching File

```
;here is an example of several patterns
EcoRI(no_spaces)        GAATTC  0  rest of line out here ignored by program
EcoRI*(N^AATTC)         NAATTC  0  at least one space after cut position
GAANN^NNTTC(XmnI)       GAANNNNTTC 4 the 4 is used to calculate fragment sizes
SOSBOX                  CTGNNNNNNNNNNCAG  find repressor binding site
```

the ambiguous bases listed in Table 2. Finally, the third field contains a number which represents the position in the sequence that would be cut by a restriction endonuclease with respect to the beginning of the pattern. (This field may be blank in which case a number zero is assumed). Then the line must have at least one space after the cut position before ending with a carriage return. The entire rest of the line beyond this space can be used for comments. The example in Table 4 illustrates all these features. Up to 150 patterns may be in the pattern matching table allowing a very extensive number of searches at the same time.

Patterns Entered from Terminal. Instead of preparing a table of patterns to be searched before using the pattern matching program, patterns may be entered as the program is running using pattern matching option 3. A pattern of up to 25 ambiguous base characters (listed in Table 2), upper or lower case, no spaces between them, is first entered, followed by a carriage return. Then a pattern name of up to 25 characters with no spaces between and followed by a carriage return is entered. Entering a carriage return for the pattern name will make the pattern itself the pattern name. There is then a prompt for the distance in nucleotides from the beginning of the matched pattern to the cut site. A return is interpreted as 0. When finished with entries, hitting an extra carriage return exits the pattern entry mode.

Approximate Matches to Patterns. It is also possible to look for an approximate match (allowing a certain number of mismatches) to a probe or a consensus regulatory sequence (pattern matching option 8). At present, this latter procedure is quite slow and does not find approximate matches involving additions or deletions.

Pattern Output Options.

Once the patterns to be matched have been selected, a variety of output options are available in the pattern output option table for displaying results. The entire sequence may be displayed with bases numbered and pattern name shown below each line (option 1, pattern output menu). The first character in the pattern name is placed directly below the first base matched in the sequence. This option also shows translation in all 3 reading frames listed above the sequence. Alternatively, only lines with matches can be displayed, a useful option for avoiding large lists of data resulting from analysis of long sequences (option 2). After the sequence lines have been shown, a table is given which shows pattern matched and base number matched in sequence. Patterns not matched are also listed in a separate table. Pattern output option 3 prints only these tables without showing any sequence. Fragment lengths can also be determined (option 4); these lengths should exactly match the 5' ends of cut sites in the given DNA strand. Note that for correct fragment lengths a third entry is necessary in the pattern matching table (see Table 4) or in entering patterns from the keyboard. The program predicts the lengths of fragments generated by digestion of both linear and circular molecules with a single restriction endonucleases.

TRANSLATION OPTIONS

Using option 1 of the translation menu (Table 1), translation can be initiated in any base range in the forward direction on the given strand or on its complement in the reverse direction. Any translation combination of the six possible reading frames is possible. For ease in identifying open reading frames, termination codons are indicated by a '*', and MET is capitalized. The sequence is printed 3 bases at a time, numbered and the translation products are given underneath. Either 1- or 3-letter amino acid code may be used.

Using translation option 2, there is a search for open reading frames (orf's) in either linear or circular sequences. Thus, if an orf starts near the end of the sequence for a circular molecule, the program will attempt to extend the orf using the first part of the sequence over again. Shorter, less significant orf's may be avoided using a minimal length option. Since the program cannot identify the start point for translation, the ribosomal binding site on the corresponding RNA, the 5' end of the sequence and all methionine codons can indicate the start of an orf. The end of the orf is the first termination codon or the 3' end, whichever occurs first. If introns are present in the sequence, the ends of the orf may not be identified correctly. To allow for this case, the program also estimates the extent of the orf by also considering a possible 5' end at the previous termination codon. The orf finding program works only on the given DNA sequence. For searching the complementary strand, option 5 of the edit menu must first be used to generate the complementary strand.

Translation option 3 uses as input a DNA sequence file previously loaded into memory using the option S, main menu, to generate a disk file with the protein sequence in standard 1- or 3-letter code. The 1-letter code file may be used directly with the Lipman-Pearson database search program dfastp (12). Option 4 uses as input a 3-letter amino acid file and back-translates it into a DNA file, using the ambiguous base symbols listed in Table 2. In an ambiguous sequence, we have assigned 3 letter codons to represent termination (ZZZ), serine(EEE), leucine(LLL) and arginine(RRR) codons. All other ambiguous bases are translated, if possible. The pattern matching program interprets these letters as representing all the possible codons for these highly degenerate amino acids. The purpose of this feature is to allow searching for silent restriction sites in coding sequences (4). Option 5 plots amino acids horizontally across the terminal screen according to hydrophobicity, charge, and other chemical features of the amino acids. The option is useful for finding possible hydrophobic regions and repeated patterns of amino acids. Options 6 provides the capability of translating a disk file containing a 1-letter amino acid code sequence into a new amino acid sequence file in 3-letter code. Option 7 provides the opposite capability, by generating a 1-letter code amino acid sequence file from an input 3-letter amino acid code file.

Format for Protein Sequence.

To enter an amino acid sequence using a text editor, the format should be similar to that of the DNA sequences. Lines with ; or > in column 1 are comments. The amino acids should be in standard 3-letter or 1-letter code, each separated from the next by one or more spaces, tabs or lines. Ter is used for termination in 3-letter code sequences and Z in 1-letter sequences. Lines can be any length and can be in upper or lower case.

Reading a Protein Sequence Diskfile. Protein sequence files in 1- or 3-letter amino acid code are treated just like DNA sequence files. Sequences are checked for incorrect entries as they are read into memory. A 3-letter code sequence must be correct or the file reading program aborts. Wrong entries in 1-letter code sequences are reported but used. The maximum length of a protein sequence which can be analyzed is usually 10,000 residues.

NUCLEOTIDE FREQUENCY ANALYSIS

Nucleotide, dinucleotide and codon usage frequencies in a DNA sequence previously loaded into memory can be analyzed using option C, main menu. As with the other programs, a range of bases can be analyzed. Nucleotides are counted starting at the lowest numbered base specified. Dinucleotide frequencies can be measured either consecutively throughout the sequence, or intermittantly by reading the first dinucleotide and then skipping the next two, thereby allowing a determination of dinucleotide frequencies in specific pairs of bases in coding regions. Codon usage results are displayed in a standard genetic code table. Ambiguous nucleotides are not scored.

EDIT SEQUENCE OPTIONS

Most of the programs included in the edit menu change an existing disk file of a sequence into some other form and place the result in a second disk file. A sequence may be displayed on the screen with numbers (option 1), a fragment may be deleted from a sequence file and placed in another disk file (option 2), a sequence may be used after a portion has been deleted (option 3), sequences in 2 different disk files may be merged into one sequence in a third disk file (option 4), a complementary sequence file may be made (option 5), a sequence file may be circularized by adding a 2 at the end (option 7) such as required to represent a circular plasmid after simulation of a cloning reaction using option 4, or a disk file containing a circular sequence may be rearranged so that the sequence starts at a different nucleotide (option 8), such as required to simulate cloning of a DNA fragment into a circular plasmid. This procedure is necessary because option 4 can only join sequences from 2 disk files end-to-end.

Option 6 of the edit menu finds the first exact match of a specified length of nucleotides in two sequences. The objective of this program is to assist in finding overlaps in two sequences quickly, such as when attempting to combine sequences from shotgun cloning experiments. This program is presently being expanded to allow for insertions and deletions as well as for mismatches in any overlap that may exist.

HOMOLOGY AND SYMMETRY ANALYSIS

We have described two programs (2) which provide a dot matrix analysis of 2 DNA sequences. The first (BASEPLOT) is a simpler one which should work on any printer. The second (SEQHOM) is a more sophisticated program which requires input as to the data which must be sent to the target printer for graphic displays. The programs PROTPLOT and PROTHOM are for corresponding homology comparisons of protein sequences on standard and dot matrix printers, respectively. These programs have been enhanced to handle very long sequences (10000-residue proteins and 480000-residue nucleic acids), to use ambiguous DNA base symbols and to be customizable to any printer.

Printer customization is achieved using a file called printer.dat, which carries information as to special codes for each printer that can be adapted by the user. Presently, SEQHOM and PROTHOM do not work with IBM and Epson type printers.

These programs display a portion of one sequence with every tenth position marked across the top and bottom of the page and a similarly marked portion of the other sequence down the sides of the page. Where two residues are the same, a dot or symbol is placed at the intersection of the row and column containing the matched residues. A window matching option restricts printing of residues to only those which are at the 5' end of a given extent of homology. This option reduces the background printing of random matches when the two sequences are compared one residue at a time. The printing of a diagonal row of residues indicates homology. Any sequence may be compared to itself to reveal the presence of direct repeats. A DNA sequence may be compared to its reversed complement to visualize dyad symmetry (13). In the above cases, the program moves sequentially through the sequences for the range of residues specified by the user, automatically advancing to the next page as each one is filled.

ADDITIONAL PROGRAMS

Three other programs from the laboratory include two that allow nucleic acid (BASECOL) and protein (COLOR) sequences to be shown in colors on the standard IBM color terminal according to chemical properties of the individual residues, and a program called THIRD that will place every third consecutive nucleotide from a sequence into a new sequence file. The latter program is useful for comparing first, second or third bases in coding regions by a dot matrix analysis (5).

ACKNOWLEDGEMENTS

D. Mount is grateful to H. Martinez for use of the VAX11/750 at UCSF. This research was supported by grants from NSF and NIH.

REFERENCES

1. Conrad, B. and Mount, D. W. (1982) Nucleic Acids Res. 10, 31-38.
2. Mount, D. W. and Conrad, B. (1984) Nucleic Acids Res. 12, part 2, 811-818.
3. Mount, D.W. and Conrad, B. (1984) Nucleic Acids Res. 12, part 2, 819-824.
4. Little, J. and Mount, D.W. (1984) Gene 32, 67-73.
5. Mount, D. (1985) BioTechniques 3, 102-112.

6. Martinez, H. (1984) Mathematical and computational problems in the analysis of molecular sequences - a special commemorative issue honoring Margaret Oakley Dayhoff. Bull. of Math. Biol. 46, No. 4.
7. Sankoff, D. and J.B. Kruskal. eds. 1984. Time warps, string edits, and macromolecules: the theory and practice of sequence comparison. Addison-Wesley Pub@. Co@., Inc@., Reading, Mass.
8. Soll, D. and R.J. Roberts, eds. 1982. The applications of computers to research on nucleic acids, I. Nucl. Acids Res. 10, no. 1.
9. Soll, D. and R.J. Roberts, eds. 1984. The applications of computers to research on nucleic acids, II. Nucl. Acids Res. 12, no. 1.
10. Jungck, J.R. and R.M. Friedman. (1984) Bull. of Math. Biol. 46, 699-744.
11. Korn, L.J. and C.L. Queen. (1984) DNA 3, 421-436.
12. Lipman, D.J. and W.R. Pearson. (1985) Science 227, 1435-1441.
13. Mount, D. Bio/Technology 2:791-795, 1984